

Integrating ArcIMS and ArcWeb for Developers With .NET

By Ismael Chivite, ESRI Application Prototype Specialist.

This article discusses how ArcIMS Image Services and the ArcWeb for Developer's Map Image service (referred to hereafter as Map Image) can be integrated. The information provided here will help a developer take advantage of existing Internet map services for building applications in a distributed environment. It is assumed that the reader is already familiar with ArcWeb for Developers, ArcIMS, and C#. A list of resources at the end of the article provides additional information.

Why Fuse?

The proliferation of Internet map services has simplified access to geographic data. Data providers publish vast and up-to-date geographic content that anyone connected to the Internet can consume. The ability to combine information from several map services to create totally new content makes one of the most interesting types of applications possible.

A common scenario for this type of data fusion would be an organization that uses third party map services as a background or reference layer for its own geographic information. For instance, customer locations or

traffic events can be viewed on top of a street or thematic map published by a data provider. In this way, an organization does not have to worry about hosting or maintaining data that is not directly related to its business.

Applying partial transparency to all colors in the foreground image might be required for an application that, for example, overlays weather information (e.g., cloud cover) on a base map. By applying a 50 percent transparency effect to all the colors in the foreground image, the basemap image will be visible through the cloud cover information. The code in Listing 3

Continued on page 42

LISTING 1

```
private Image DownloadImageFromURL (string sImageURL)
{
    System.Net.WebClient pWebClient = new System.Net.WebClient();
    return Image.FromStream(pWebClient.OpenRead (sImageURL));
}
```

LISTING 2

```
private Image FuseImages (Image BGImg, Image FGImg, Color TransparentColor)
{
    //Create the ColorMap for setting transparency to the foreground Image
    ImageAttributes pImgAtt = new ImageAttributes();
    ColorMap myColorMap = new ColorMap();
    myColorMap.OldColor = TransparentColor;
    myColorMap.NewColor = Color.FromArgb(0,0,0,0);
    System.Drawing.Imaging.ColorMap[] remapTable = new ColorMap[1];
    remapTable[0] = myColorMap;

    pImgAtt.SetRemapTable(remapTable, ColorAdjustType.Bitmap);

    //Create a new image to draw to.
    Image compositeImage = new Bitmap(BGImg.Width, BGImg.Height, PixelFormat.Format64bppArgb);
    Graphics compGrphcs = Graphics.FromImage(compositeImage);

    //Drawing BackGround Image
    compGrphcs.DrawImage (BGImg, new Rectangle(0,0, BGImg.Width, BGImg.Height), 0, 0,
    BGImg.Width, BGImg.Height, GraphicsUnit.Pixel);
    //Drawing ForeGround Image (transparency effect)
    compGrphcs.DrawImage(FGImg, new Rectangle(0,0, FGImg.Width, FGImg.Height), 0, 0,
    FGImg.Width, FGImg.Height, GraphicsUnit.Pixel, pImgAtt);

    return compositeImage;
}
```

Integrating ArcIMS and ArcWeb for Developers With .NET

Continued from page 41

fuses two images setting a translucency effect to the foreground image.

The ArcWeb for Developers' Map Image Service

Part of ArcWeb for Developers, ArcWeb Services is a suite of Simple Object Access Protocol (SOAP)-based Web services offering GIS functionality. Specifically, the Map Image service publishes aerial imagery, street, thematic, and topographic maps. An exhaustive list of the available data sources is included in an ESRI white paper entitled *An Overview of ArcWeb for Developers* that can be downloaded from the ESRI Web site. The examples in this article use ArcWeb for Developers as a source for reference maps.

ArcIMS Image Services

ArcIMS offers a powerful set of tools for publishing maps over the Internet. Hundreds of public ArcIMS services that provide geographic data via the Internet can be located using the Geography Network. For the purposes of this article, it is assumed that an ArcIMS service publishes business information will be drawn on top of the reference map provided by ArcWeb for Developers.

Requesting Images

ArcWeb for Developers provides an easy mechanism for requesting map images. A SOAP Application Program Interface (API) can be used to specify the extent, data source, size, and other options for the requested map.

Take Advantage of Multithreading

When requesting two or more images, start a new thread for each Internet map service and wait for all of them to respond. Response times will improve because the Internet Map Services will run in parallel.

Map images from an existing ArcIMS service can be accessed through a powerful XML API (i.e., ArcXML) that allows the developer to specify even more properties for the map being retrieved. In either case, the maps requested are generated by the server as images that client applications can download and use. Since this article is concerned only with the process of combining an ArcIMS Image Ser-

LISTING 3

```
private Image FuseImages (Image BGImg, Image FGImg, double TranspLevel)
{
    // TranspLevel is a value from 0 to 1

    if (TranspLevel == 0) return FGImg;
    if (TranspLevel == 1) return BGImg;

    //Create the color matrix
    ImageAttributes pImgAtt = new ImageAttributes();
    float[][] colorMatrixElements =
    {
        new float[] {1.0f, 0.0f, 0.0f, 0.0f, 0.0f},
        new float[] {0.0f, 1.0f, 0.0f, 0.0f, 0.0f},
        new float[] {0.0f, 0.0f, 1.0f, 0.0f, 0.0f},
        new float[] {0.0f, 0.0f, 0.0f, Convert.ToSingle(TranspLevel), 0.0f},
        new float[] {0.0f, 0.0f, 0.0f, 0.0f, 1.0f}
    };

    ColorMatrix wmColorMatrix = new ColorMatrix(colorMatrixElements);

    pImgAtt.SetColorMatrix(wmColorMatrix, ColorMatrixFlag.Default,
        ColorAdjustType.Bitmap);

    //Superimpose both Images
    Image compositeImage = new Bitmap(BGImg.Width, BGImg.Height,
    PixelFormat.Format64bppPArgb);
    Graphics compGrphcs = Graphics.FromImage(compositeImage);

    //Drawing Background Image
    compGrphcs.DrawImage (BGImg, new rectangle(0,0,BGImg.Width,BGImg.Height),0,0,
    BGImg.Width,BGImg.Height,GraphicsUnit.Pixel);
    //Drawing ForeGround Image (translucid effect)
    compGrphcs.DrawImage(FGImg, new Rectangle(0,0,FGImg.Width,FGImg.Height),0,0,
    FGImg.Width,FGImg.Height,GraphicsUnit.Pixel,pImgAtt);

    return compositeImage;
}
```

vices and Map Image, it is assumed that the images have already been requested.

Accessing Images

The simple .NET function shown in Listing 1 can be used to access the images created by each service. Note that the accessed images are not saved to disk but kept as a System.Drawing.Image object for later use.

Fusing Images

When overlaying two images, the foreground image requires some transparency setting so that the reference data can be seen. The two most common transparency effects are setting a background color as transparent in the foreground image and applying partial transparency to all colors in the foreground image.

The first option—setting a background color as transparent in the foreground image—is generally used when labeled points or polylines must be drawn on top of the background image. If a street map from ArcWeb for Developers can be used as a background for customer locations, traffic events, or other data that was published using ArcIMS. The code in Listing 2 fuses two images and sets a specific foreground image's color to transparent.

Caching for Improved Performance

If the content of a map service will not change during a user's session, cache and reuse the requested images requested.

Using the Combined Image

After combining both images, the image can be saved to disk or used directly. Because writing to disk is generally an expensive process, directly sending the image back (in the case of HTML clients) or setting the image property of a Windows Form Control (in case of stand alone applications) is preferred. More information on these techniques is contained in the Image Grabbers sample available from ArcScripts at arcscripts.esri.com.

Conclusion

In a distributed environment, an application can make use of different services to retrieve and combine two or more images and generate totally new content. Microsoft .NET can be used to fuse ArcWeb for Developers and ArcIMS maps into a single image by using transparency effects.

Resources

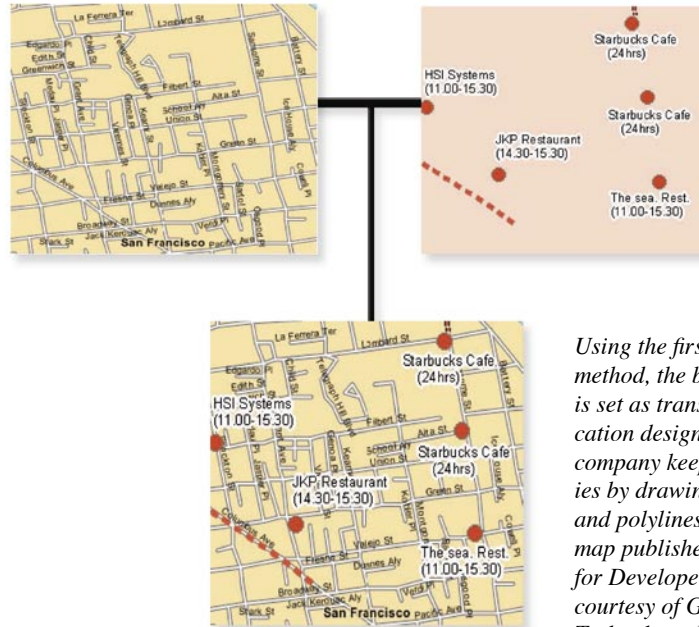
For a listing of the data sources available from ArcWeb for Developers, see the white paper *An Overview of ArcWeb for Developers* which can be downloaded from www.esri.com/library/whitepapers/pdfs/arcweb-for-developers.pdf.

For information on ArcWeb for Developers, visit arcweb.esri.com/arcwebonline/.

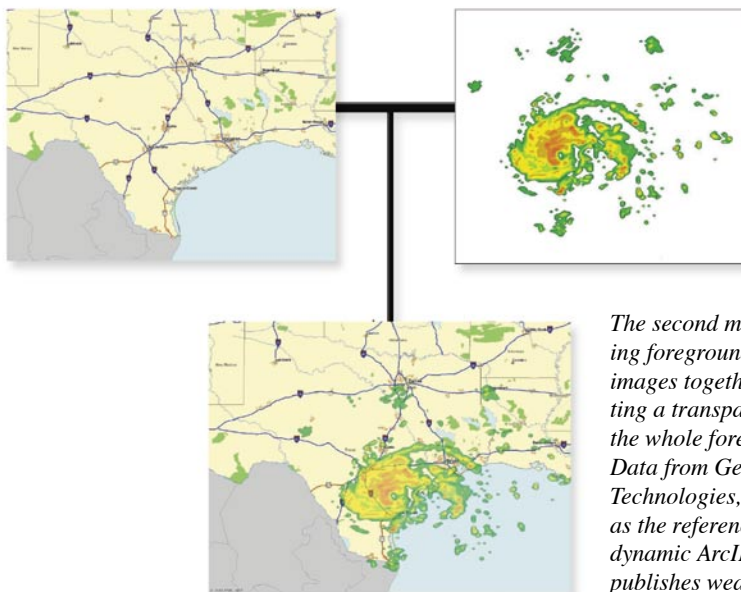
Information on the XML API

The ArcXML Programmer's Reference Guide is available online at support.esri.com.

For information on requesting images and working with combined images, see a sample Web application called Image Grabbers which can be downloaded from ArcScripts at www.esri.com/arcscripts.



Using the first image fusing method, the background color is set as transparent. An application designed for a logistics company keeps track of deliveries by drawing business points and polylines on top of a street map published through ArcWeb for Developers. Street data courtesy of Geographic Data Technology, Inc.



The second method for viewing foreground and basemap images together involves setting a transparency level for the whole foreground image. Data from Geographic Data Technologies, Inc., is used as the reference layer for a dynamic ArcIMS service that publishes weather data. A constant transparency was applied to the foreground image so data underneath can be viewed.